Journal of Network Communications and Emerging Technologies (JNCET) www.jncet.org
Volume 8, Issue 4, April (2018)

Domain Specific Image Blur Detection using Neural Networks

D. Bhaskar [1], Chinmay Kousik [2], Subham Patjoshi [3], Naina Shamdasani [4], Shruti Shreya [5]
[1]Assistant Professor, Department Of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, India.
[2,3,4,5] Department Of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, India.

Abstract – Image collection and analysis is on the rise in many commercial domains such as it is useful for processes such as quality control, security, product testing etc. In these cases, image degradation caused by environmental factors is undesirable and methods for quick identification and recovery of degraded images are required. This paper proposes a method for domain specific blur detection using wavelet transforms and neural networks.

Index Terms – Haar wavelet transform, neural networks, image processing, image recovery

1. INTRODUCTION

Image collection and processing plays a crucial role in industries as it is an important step towards the automation of many different processes. In industrial applications, image processing and machine vision may be used for counting objects, detecting flaws in manufactured items, and for taking measurements. This requires that images be collected frequently and accurately, since any significant degradation in image quality can severely affect the performance of the manufacturing process. Therefore, image degradation must be detected quickly and attempts must be made to recover a visually plausible sharp image from the degraded image.

The factors that contribute to image degradation may be varied, such as:

- Fluctuation in lighting conditions during image collection.
- Movement of the subject or camera.
- Electronic interference from external sources.
- Defects in the camera apparatus.

A property of domain specific image collection setups is that in many cases, it can be assumed that the collector is drawing samples from a dataset with mostly consistent degradation factors. From this, it can be inferred that the degradation caused by the environmental factors in the domain under consideration will remain consistent over a period of time.

This provides a compelling case for using machine learning for classifying images as blurred or not blurred. Features from an image can be extracted and passed to a neural network for learning and classification.

In this system, feature extraction is done using Haar Wavelet Transform [5] and Laplacian Variance [6]. The extracted features are then passed to a feed forward neural network that uses a perceptron learning algorithm [7] to determine if an image is blurred or not.

2. RELATED WORK

2.1 Related Works for Blur Detection

- Blur detection for Digital Images using Wavelet Transform (ICME 2004)
- Blur detection using Laplacian Operator and Open-CV. (SMART 2016)
- Space-invariant deblurring given N independently blurred images of a common object.
- Blind deconvolution by means of the Richardson-Lucy algorithm.

2.2 Drawbacks of Existing Systems

The existing systems make assumptions that the image blur is spatially invariant. This is not true in cases where parts of the collected image may be in motion with respect to the camera, or where the lighting condition across the image is not consistent. This results in different parts of the image being blurred with different Point Spread Functions (PSFs).

The disadvantage of computing a blur coefficient using the Laplacian operator is that the operator is highly sensitive to noise in the image. Therefore denoising the image is important before convolving it with the Laplacian operator.

The disadvantage of blur detection using only wavelet transforms is that the thresholds for blur classification need to be determined using empirical methods. This is a slow and time consuming process.

In many cases, we may only have a single image of an object and using the method described in [4] may not be applicable.

3. PROPOSED MODELLING

3.1 Modeling Image Blur

A blurred image with spatially invariant blur may be modeled using the following equation.

$$B = I \quad H + N \tag{1}$$

where B is the blurred image, I is the visually plausible sharp image, and N is the noise added to the image [3]. The quantity H represents the Point Spread Function, which describes how each pixel is blurred. For a spatially invariant blur with low noise, computing the Fourier transform of the data leads to the following equation:

$$I = F^{-1}\left(\frac{F(B)}{F(H)}\right) \tag{2}$$

where F and F^{-1} denote the Fourier transform and Inverse Fourier transform respectively. Modifying the formula to handle spatially variant blur, we get:

$$I = F^{-1}\left(\frac{F(B)}{h(x,y)}\right) \tag{3}$$

where $h(x,y)$ is the Point Spread Function at pixel (x,y).

The problem with deblurring is that both I and H are unknowns, and there are infinite pairs of I and H that satisfy the above equation. Methods for deblurring images without prior knowledge of the Point Spread Function use initial guesses of the function and in subsequent iterations improve upon the result. An example of such a method is the Richardson-Lucy Blind Deconvolution Algorithm [1].

The blur may also be estimated using the sharpness of edges. The Discrete Wavelet Transform technique and Variance of Laplacian technique both use this method.

The Laplacian kernel is given by:

$$L = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \tag{4}$$

3.2 Blur Detection Method

The blur detection method consists of two algorithms; one for estimating the edges and blur coefficient list in the image, and the other for determining if an image is blurred or not using a perceptron.

3.2.1 Blur Coefficient Estimation

Perform a Discrete Wavelet Transform on the image using Haar Wavelet and decomposition level 1. Let *LL, LH, HL,* and *HH* be the resulting Low to Low, Low to High, High to Low, and High to High decomposition bands respectively, obtained from the wavelet transform. Here, *LL* provides an approximation of the input image, *LH* provides the horizontal edge components of the image, *HL* provides the vertical edge components of the image, and *HH* provides the diagonal edge components of the image.

For each pixel (x,y) in the resulting decomposition bands, compute the following:

$$Edg_{x,y} = \sqrt{LH_{x,y}^2 + HL_{x,y}^2 + HH_{x,y}^2} \tag{5}$$

Once the Edge map *Edg* has been computed, divide the image into N equally sized segments. For eg. for a *256 * 256 pixel* edge map, it may be useful to divide the image into sections of size *8 * 8*. Let the resulting list of image segments be *Seg*.

For each segment 'i' in the resulting segments compute the blur extent as follows:

$$Bl_i = \sigma^2(Seg_i \square L) \tag{6}$$

where Bl is an array of blur extent, L is the Laplacian kernel, and $X \square Y$ denotes the matrix convolution of matrices X and Y, and σ^2 denotes the variance operation.

The array Bl is returned as the result.

The resulting array Bl denotes to what extent each image segment is blurred. Since the number of segments is constant, it can be passed to a neural network as input for analysis. This allows the system to learn which images are blurred and which images are visually plausible.

3.2.2 Perceptron Training

Initialize a neural network with N inputs and a bias term. Randomize the input coefficients.

Since the elements of the input vector Bl may fall in the interval $(0, 1000)$, it is important to normalize these terms. Therefore, compute the input vector to the neural network z using the following formula:

$$z_i = \frac{\log_{10}(Bl_i)}{3} \tag{7}$$

A logarithmic function is used since high values of Bl mean that the segment is not blurred, and hence the value of z does not vary much. After normalization, the elements of z lie in the interval $(0, 1)$.

Set the activation function of the output neuron to the Sigmoid activation function given by

$$f(z_i) = \frac{1}{1+e^{z_i}} \tag{8}$$

since the output is a binary classification.

For each each image in the training data set, compute the blur extent array and tag the images as blurred or clear.

Train the model using the perceptron learning algorithm specified in [7] and the training data set.

Once the algorithm has been trained using the perceptron learning algorithm [7], it can be used to classify images as blurred or not blurred for the given dataset.

4. RESULTS AND DISCUSSIONS

Applying the edge map generation algorithm to the image shown in Figure 1 yields Figure 2.

Figure 1. Input clear image

The Discrete Wavelet Transform is not very sensitive to noise in the image and outputs a reasonably accurate picture of the important edge in the input image. This is done by combining the vertical, horizontal, and diagonal decompositions using Equation (5).

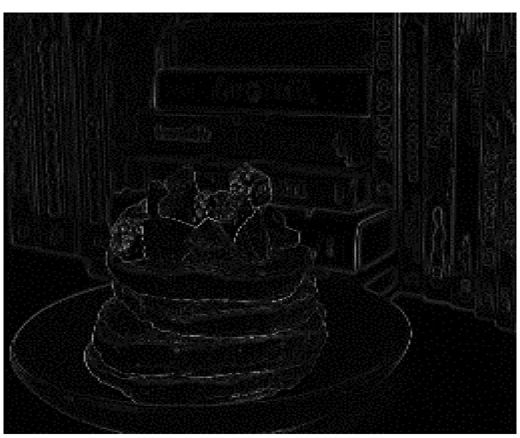


Figure 2. Resultant Edge Map of the Image

The resulting values of calculating the Laplacian variance are as follows:

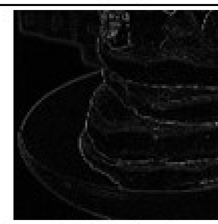
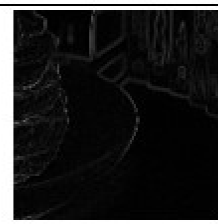
Image Section	Bl
 Top left	0.0167721843206
 Top right	0.024008201833
 Bottom Left	0.024008201833
 Bottom Right	0.0121958337637

Table 1. Blur Coefficients for Sharp Image

Applying the same algorithm to the image blurred with a *(5,5)* dimensional Gaussian blur kernel we get the following results:

Figure 3. Image Blurred with 5*5 Gaussian kernel

Figure 4. Edge Map of Figure 3

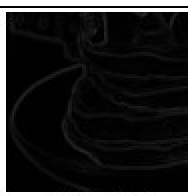
Image Section	Bl
 Top Left	0.00137363077507
 Top Right	0.0013341695311
 Bottom Left	0.00121547333055
 Bottom Right	0.00101239317055

Table 2. Blur Coefficients for Blurred image

It is shown that the variance in the sharp image is greater than the variance of the blurred image by an order of magnitude. These differences can be learned by the neural network to classify images as blurred or not blurred.

From the edge map it is clear that most of the edges of the image are quite significantly reduced in intensity. This change in intensity should reduce the total variance of the convolution of the input map section with the Laplacian kernel. The values of the output vector *Bl* are shown in Table 2.

5. CONCLUSION

aThis paper introduces a method for classifying images as blurred and not blurred from a given data set. The results are shown using a representative image. The method introduced in this paper also uses edge based techniques for feature extraction from images and uses introduces the use of a neural network for classifiction. An example use case for this method would be in automated quality assessment systems in the manufacturing sector. There is also significant work to be done in finding regions with similar Point Spread Functions without dividing the image into segments of fixed size. Using smaller sized segments can help locate areas of the image with significant blur and minimum overlap with other images at the cost of processing time.

REFERENCES

[1] D. A. Fish, A. M. Brinicombe, E. R. Pike, and J. G. Walker, Blind deconvolution by means of the RichardsonLucy algorithm, In: J. Opt. Soc. Am. A 12, 58-65 (1995)

[2] J. Pan, D. Sun, H. Pfister and M. H. Yang, Deblurring Images via Dark Channel Prior, In: IEEE Transactions on Pattern Analysis and Machine Intelligence . doi: 10.1109/TPAMI.2017.2753804

[3] R. L. Lagendijk, Basic Method for Image Restoration and identification, In: Academic Press, (2000)

[4] Dennis C. Ghiglia, Space-invariant deblurring given N independently blurred images of a common object, J. Opt. Soc. Am. A 1, 398-402 (1984)

[5] Hanghang Tong, Mingjing Li, Hongjiang Zhang and Changshui Zhang, ”Blur detection for digital images using wavelet transform,In: 2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763), (2004)

[6] R. Bansal, G. Raj and T. Choudhury, Blur image detection using Laplacian operator and Open-CV, In: 2016 International Conference System Modeling & Advancement in Research Trends (SMART), Moradabad, (2016)

[7] Freund, Yoav and Schapire, Robert E., Large Margin Classification Using the Perceptron Algorithm, In” Machine Learning (1999)